

# Scalable Nonparametric Bayesian Inference on Point Processes with Gaussian Processes

Yves-Laurent Kom Samo  
Stephen Roberts

YVES-LAURENT.KOMSAMO@ENG.OX.AC.UK  
SJROB@ROBOTS.OX.AC.UK

Department of Engineering Science and Oxford-Man Institute, University of Oxford

## Abstract

In this paper we propose an efficient, scalable non-parametric Gaussian Process model for inference on Poisson Point Processes. Our model does not resort to gridding the domain or to introducing latent *thinning* points. Unlike competing models that scale as  $\mathcal{O}(n^3)$  over  $n$  data points, our model has a complexity  $\mathcal{O}(nk^2)$  where  $k \ll n$ . We propose a MCMC sampler and show that the model obtained is faster, more accurate and generates less correlated samples than competing approaches on both synthetic and real-life data. Finally, we show that our model easily handles data sizes not considered thus far by alternate approaches.

## 1. INTRODUCTION

Point processes are a standard model when the objects of study are the number and repartition of otherwise identical points on a domain, usually time or space. The Poisson Point Process is probably the most commonly used point process. It is fully characterised by an intensity function that is inferred from the data. Gaussian Processes have been successfully used to form a prior over the (log-) intensity function for applications such as astronomy (Gregory & Lored, 1992), forestry (Heikkinen & Arjas, 1999), finance (Basu & Dassios, 2002), and neuroscience (Cunningham et al., 2008b). We offer extensions to existing work as follows: we develop an exact non-parametric Bayesian model that enables inference on Poisson processes. Our method scales linearly with the number of data points and does not resort to gridding the domain. We derive a MCMC sampler for core components of the model and show that our approach offers a faster and more accurate solution, as well as producing less correlated samples,

compared to other approaches on both real-life and synthetic data.

## 2. RELATED WORK

Non-parametric inference on point processes has been extensively studied in the literature. (Rathbun & Cressie, 1994) and (Moeller et al., 1998) used a finite-dimensional piecewise constant log-Gaussian for the intensity function. Such approximations are limited in that the choice of the grid on which to represent the intensity function is arbitrary and one has to trade-off precision with computational complexity and numerical accuracy, with the complexity being cubic in the precision and exponential in the dimension of the input space. (Kottas, 2006; Kottas & Sanso, 2007) used a Dirichlet process mixture of Beta distributions as prior for the normalised intensity function of a Poisson process. (Cunningham et al., 2008a) proposed a model using Gaussian Processes evaluated on a fixed grid for the estimation of intensity functions of renewal processes with log-concave renewal distributions. They turned hyper-parameters inference into an iterative series of convex optimization problems, where ordinarily cubic complexity operations such as Cholesky decompositions are evaluated in  $\mathcal{O}(n \log n)$  leveraging the uniformity of the grid and the log-concavity of the renewal distribution. (Adams et al., 2009) proposed an exact Markov Chain Monte Carlo (MCMC) inference scheme for the posterior intensity function of a Poisson Process with a Sigmoid Gaussian prior intensity, or equivalently a Cox Process (Cox, 1955) with Sigmoid Gaussian stochastic intensity. The authors simplified the likelihood of a Cox process by introducing latent *thinning points*. The proposed scheme has a complexity exponential in the dimension of the input space, cubic in the number of data and thinning points, and performs particularly poorly when the data are sparse. (Gunter et al., 2014) extended this model to structured point processes. (Rao & Teh, 2011) used *uniformization* to produce exact samples from a non-stationary renewal process whose hazard function is modulated by a Gaussian Process, and consequently proposed an MCMC

sampler to sample from the posterior intensity of a unidimensional point process. Although the authors have illustrated that their model is faster than (Adams et al., 2009) on some synthetic and real-life data, their method still scales cubically in the number of thinned and data points, and is not applicable to data in dimension higher than 1, such as spatial point processes.

### 3. MODEL

#### 3.1. Setup

We are tasked with making non-parametric Bayesian inference on the intensity function of a Poisson Point Process assumed to have generated a dataset  $\mathcal{D} = \{s_1, \dots, s_n\}$ . To simplify the discourse without loss of generality, we will assume that data points take values in  $\mathbb{R}^d$ .

Firstly, let us recall that a **Poisson Point Process** (PPP) on a bounded domain  $\mathcal{S} \subset \mathbb{R}^d$  with non-negative **intensity function**  $\lambda$  is a locally finite random collection of points in  $\mathcal{S}$  such that the numbers of points occurring in disjoint parts  $B_i$  of  $\mathcal{S}$  are independent and each follows a Poisson distribution with mean  $\int_{B_i} \lambda(s) ds$ .

The likelihood of a PPP is given by:

$$L(\lambda | s_1, \dots, s_n) = \exp \left( - \int_{\mathcal{S}} \lambda(s) ds \right) \prod_{i=1}^n \lambda(s_i) \quad (1)$$

#### 3.2. Tractability discussion

The approach adopted thus far in the literature to make non-parametric Bayesian inference on Point Process using Gaussian Processes (GP) (Rasmussen & Williams, 2006) consists of putting a *functional prior* on the intensity function in the form of a positive function of a GP:  $\lambda(s) = f(g(s))$  where  $g$  is drawn from a GP and  $f$  is a positive function. Examples of such  $f$  include the exponential function and a scaled sigmoid function (Adams et al., 2009; Rao & Teh, 2011). This approach can be seen as a Cox Process where the stochastic intensity follows the same dynamics as the functional prior. When the Gaussian Process used has almost surely continuous paths, the random vector

$$(\lambda(s_1), \dots, \lambda(s_n), \int_{\mathcal{S}} \lambda(s) ds) \quad (2)$$

provably admits a probability density function (pdf). Moreover, we note that any piece of information not contained in the implied pdf over the vector in Equation (2) will be lost as the likelihood only depends on those variables. Hence, given a functional prior postulated on the intensity function, the only necessary piece of information to be able to make a full Bayesian treatment is the implied joint pdf over the vector in Equation (2).

For many useful transformations  $f$  and covariance structures for the GP, the aforementioned implied pdf might not be available analytically. We note however that there is no need to put a *functional prior* on the intensity function. In fact, for every *finite-dimensional prior* over the vector in Equation (2), there exists a Cox process with an a.s.  $C^\infty$  intensity process that coincides with the postulated prior (see appendix for the proof).

This approach is similar to that of (Kottas, 2006). The author regarded  $I = \int_{\mathcal{S}} \lambda(s) ds$  as a random variable and noted that  $p(s) = \frac{\lambda(s)}{\int_{\mathcal{S}} \lambda(s) ds}$  can be regarded as a pdf whose support is the domain  $\mathcal{S}$ . He then made inference on  $(I, p(s_1), \dots, p(s_n))$ , postulating as prior that  $I$  and  $(p(s_1), \dots, p(s_n))$  are independent,  $I$  has a Jeffreys prior and  $(s_1, \dots, s_n)$  are i.i.d. draws from a Dirichlet Process mixture of Beta with pdf  $p$ .

The model we present in the following section puts an appropriate *finite-dimensional prior* on  $(\lambda(s_1), \dots, \lambda(s_n), \lambda(s'_1), \dots, \lambda(s'_k), \int_{\mathcal{S}} \lambda(s) ds)$  for some inducing points  $s'_j$  rather than putting a *functional prior* on the intensity function directly.

#### 3.3. Our model

##### 3.3.1. INTUITION

The intuition behind our model is that the data are not a ‘natural grid’ at which to infer the value of the intensity function. For instance, if the data consists of 200,000 points on the interval  $[0, 24]$  as in one of our experiments, it might not be necessary to infer the value of a function at 200,000 points to characterise it on  $[0, 24]$ . Instead, we find a small set of inducing points  $\mathcal{D}' = \{s'_1, \dots, s'_k\}$ ,  $k \ll n$  on our domain, through which we will define the prior over the vector in Equation (2) augmented with  $\lambda(s'_1), \dots, \lambda(s'_k)$ . The set of inducing points will be chosen so that knowing  $\lambda(s'_1), \dots, \lambda(s'_k)$  would result in knowing the values of the intensity function elsewhere on the domain, in particular  $\lambda(s_1), \dots, \lambda(s_n)$ , with ‘arbitrary certainty’. We will then analytically integrate out the dependency in  $\lambda(s_1), \dots, \lambda(s_n)$  from the posterior, thereby reducing the complexity from cubic to linear in the number of data points without ‘loss of information’, and reformulating our problem as that of making exact Bayesian inference on the value of the intensity function at the inducing points. We will then describe how to obtain predictive mean and variance of the intensity function elsewhere on the domain from training.

##### 3.3.2. MODEL SPECIFICATION

Let us denote by  $\lambda^*$  a positive stochastic process on  $\mathcal{S}$  such that  $\log \lambda^*$  is a stationary Gaussian Process with covariance kernel  $\gamma^* : (s_1, s_2) \rightarrow \gamma^*(s_1, s_2)$  and constant

mean  $m^*$ . Let us further denote by  $\hat{\lambda}$  a positive stochastic process on  $\mathcal{S}$  such that  $\log \hat{\lambda}$  is a **Conditional Gaussian Process** coinciding with  $\log \lambda^*$  at  $k$  inducing points  $\mathcal{D}' = \{s'_1, \dots, s'_k\}$ ,  $k \ll n$ . That is,  $\log \hat{\lambda}$  is the non-stationary Gaussian Process whose mean function  $m$  is defined by

$$m(s) = m^* + \Sigma_{\mathcal{D}'}^* \Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1} G \quad (3)$$

where  $G = (\log \lambda^*(s'_1) - m^*, \dots, \log \lambda^*(s'_k) - m^*)$  and  $\Sigma_{XY}^*$  is the covariance matrix between the vectors  $X$  and  $Y$  under the covariance kernel  $\gamma^*$ . Moreover,  $\log \hat{\lambda}$  is such that for every vector  $S_1$  of points in  $\mathcal{S}$ , the auto-covariance matrix  $\Sigma_{S_1 S_1}$  of the values of process at  $S_1$  reads<sup>1</sup>

$$\Sigma_{S_1 S_1} = \Sigma_{S_1 S_1}^* - \Sigma_{S_1 \mathcal{D}'}^* \Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1} \Sigma_{S_1 \mathcal{D}'}^{*T}. \quad (4)$$

The prior distribution in our model is constructed as follows:

1.  $\{\log \lambda(s'_i)\}_{i=1}^k$  are samples from the stationary GP  $\log \lambda^*$  at  $\{s'_i\}_{i=1}^k$  respectively, with  $m^* = \log \frac{\#\mathcal{D}}{\mu(\mathcal{S})}$ , where  $\mu(\mathcal{S})$  is the size of the domain.
2.  $I = \int_{\mathcal{S}} \lambda(s) ds$  and  $\{\log \lambda(s_j)\}_{j=1}^n$  are conditionally independent given  $\{\log \lambda(s'_i)\}_{i=1}^k$ .
3. Conditional on  $\{\log \lambda(s'_i)\}_{i=1}^k$ ,  $\{\log \lambda(s_j)\}_{j=1}^n$  are independent, and for each  $j \in [1..n]$   $\log \lambda(s_j)$  follows the same distribution as  $\log \hat{\lambda}(s_j)$ .
4. Conditional on  $\{\log \lambda(s'_i)\}_{i=1}^k$ ,  $I$  follows a Gamma distribution with shape  $\alpha_I$  and scale  $\beta_I$ .
5. The mean  $\mu_I = \alpha_I \beta_I$  and variance  $\sigma_I^2 = \alpha_I \beta_I^2$  of  $I$  are that of  $\int_{\mathcal{S}} \hat{\lambda}(s) ds$ .

Assertion 3. above is somewhat similar to the FITC model of (Quinero & Rasmussen, 2005).

This construction yields a prior pdf of the form:

$$\begin{aligned} & p(\log \lambda(s_1), \dots, \log \lambda(s_n), \log \lambda(s'_1), \dots, \log \lambda(s'_k), I, \theta) \\ &= \mathcal{N}(\log \lambda(s'_1), \dots, \log \lambda(s'_k) | m^* 1_k, \Sigma_{\mathcal{D}'\mathcal{D}'}^*) \\ &\times \mathcal{N}(\log \lambda(s_1), \dots, \log \lambda(s_n) | M, \text{diag}(\Sigma_{\mathcal{D}\mathcal{D}})) \\ &\times \gamma_d(I | \alpha_I, \beta_I) \times p(\theta) \end{aligned} \quad (5)$$

where  $\mathcal{N}(\cdot | X, C)$  is the multivariate Gaussian pdf with mean  $X$  and covariance matrix  $C$ ,  $M = (m(s_1), \dots, m(s_n))$ ,  $1_k$  is the vector with length  $k$  and elements 1,  $\text{diag}(\Sigma_{\mathcal{D}\mathcal{D}})$  is the diagonal matrix whose diagonal

<sup>1</sup>The positive definiteness of the induced covariance kernel  $\gamma$  is a direct consequence of the positive definiteness of  $\gamma^*$ .

is that of  $\Sigma_{\mathcal{D}\mathcal{D}}$ ,  $\gamma_d(x | \alpha, \beta)$  is the pdf of the gamma distribution with shape  $\alpha$  and scale  $\beta$ , and where  $\theta$  denotes the hyper-parameters of the covariance kernel  $\gamma^*$ .

It follows from the fifth assertion in our prior specification that  $\alpha_I = \frac{\mu_I^2}{\sigma_I^2}$  and  $\beta_I = \frac{\sigma_I^2}{\mu_I}$ . We also note that

$$\begin{aligned} \mu_I &= \mathbb{E} \left( \int_{\mathcal{S}} \hat{\lambda}(s) ds \right) \\ &= \int_{\mathcal{S}} \mathbb{E}(\exp(\log \hat{\lambda}(s))) ds \\ &= \int_{\mathcal{S}} \exp(m(s) + \frac{1}{2} \gamma(s, s)) ds := \int_{\mathcal{S}} f(s) ds \end{aligned} \quad (6)$$

and

$$\begin{aligned} \sigma_I^2 &= \mathbb{E} \left( \left( \int_{\mathcal{S}} \hat{\lambda}(s) ds \right)^2 \right) - \mu_I^2 \\ &= \mathbb{E} \left( \int_{\mathcal{S}} \int_{\mathcal{S}} \exp(\log \hat{\lambda}(s_1) + \log \hat{\lambda}(s_2)) ds_1 ds_2 \right) - \mu_I^2 \\ &= \int_{\mathcal{S}} \int_{\mathcal{S}} \mathbb{E} \left( \exp(\log \hat{\lambda}(s_1) + \log \hat{\lambda}(s_2)) \right) ds_1 ds_2 - \mu_I^2 \\ &= \int_{\mathcal{S}} \int_{\mathcal{S}} \exp(m(s_1) + m(s_2) + \gamma(s_1, s_2) + \frac{1}{2} \gamma(s_1, s_1) \\ &\quad + \frac{1}{2} \gamma(s_2, s_2)) ds_1 ds_2 - \mu_I^2 \\ &:= \int_{\mathcal{S}} \int_{\mathcal{S}} g(s_1, s_2) ds_1 ds_2 - \mu_I^2. \end{aligned} \quad (7)$$

The integrals in Equations (6) and (7) can be easily evaluated with numerical methods such as Gauss-Legendre quadrature (Hildebrand, 2003).

In particular, when  $\mathcal{S} = [a, b]$ ,

$$\mu_I \approx \frac{b-a}{2} \sum_{i=1}^p \omega_i f \left( \frac{b-a}{2} x_i + \frac{b+a}{2} \right) \quad (8)$$

and

$$\begin{aligned} \sigma_I^2 &\approx \frac{(b-a)^2}{4} \sum_{i=1}^p \sum_{j=1}^p \omega_i \omega_j g \left( \frac{b-a}{2} x_i + \frac{b+a}{2}, \frac{b-a}{2} x_j + \right. \\ &\quad \left. \frac{b+a}{2} \right) - \mu_I^2 \end{aligned} \quad (9)$$

where the roots  $x_i$  of the Legendre polynomial of order  $p$  and the weights  $\omega_i$  are readily available from standard textbooks on numerical analysis such as (Hildebrand, 2003) and scientific programming packages (R, Matlab and Scipy). Extensions to rectangles in higher dimensions are straightforward. Moreover, the complexity of such approximations only depends on the number of inducing points and  $p$  (see Equations (3) and (4)), and hence *scales well* with the data size.

A critical step in the derivation of our model is to analytically integrate out  $\log \lambda(s_1), \dots, \log \lambda(s_n)$  in the posterior, to eliminate the cubic complexity in the number of data points. To do so, we note that:

$$\begin{aligned} & \int_{(\mathbb{R}^d)^n} \prod_{i=1}^n \lambda(s_i) \mathcal{N}(\log \lambda(s_1), \dots, \log \lambda(s_n) | M, \\ & \text{diag}(\Sigma_{\mathcal{D}\mathcal{D}})) d \log \lambda(s_1) \dots d \log \lambda(s_n) \\ &= \mathbb{E} \left( \exp \left( \sum_{i=1}^n \log \lambda(s_i) \right) \right) \\ &= \exp(1_n^T M + \frac{1}{2} \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}})) \end{aligned} \quad (10)$$

where the second equality results from the moment generating function of a multivariate Gaussian.

Thus, putting together the likelihood of Equation (1) and Equation (5), and integrating out  $(\log \lambda(s_1), \dots, \log \lambda(s_n))$ , we get:

$$\begin{aligned} & p(\log \lambda(s'_1), \dots, \log \lambda(s'_k), I, \theta | \mathcal{D}) \\ & \sim p(\theta) \mathcal{N}(\log \lambda(s'_1), \dots, \log \lambda(s'_k) | M^*, \Sigma_{\mathcal{D}'\mathcal{D}'}^*) \\ & \times \exp(1_n^T M + \frac{1}{2} \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}})) \exp(-I) \gamma_d(I | \alpha_I, \beta_I) \end{aligned} \quad (11)$$

Finally, although our model allows for joint inference on the intensity function and its integral, we restrict our attention to making inference on the intensity function for brevity. By integrating out  $I$  from Equation (11), we get the new posterior:

$$\begin{aligned} & p(\lambda, \theta | \mathcal{D}) := p(\log \lambda(s'_1), \dots, \log \lambda(s'_k), \theta | \mathcal{D}) \quad (12) \\ & \sim p(\theta) \exp(1_n^T M + \frac{1}{2} \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}})) (1 + \beta_I)^{-\alpha_I} \\ & \times \mathcal{N}(\log \lambda(s'_1), \dots, \log \lambda(s'_k) | M^*, \Sigma_{\mathcal{D}'\mathcal{D}'}^*) \end{aligned}$$

where we noted that the dependencies of Equation (11) in  $I$  is of the form  $\exp(-x) \gamma_d(x | \alpha, \beta)$  which can be integrated out as the moment generating function of the gamma distribution evaluated at  $-1$ , that is  $(1 + \beta)^{-\alpha}$ .

### 3.3.3. SELECTION OF INDUCING POINTS

Inferring the number  $k$  and positions of the inducing points  $s'_i$  is critical to our model, as  $k$  directly affects the complexity of our scheme and the positions of the inducing points affect the quality of our prediction. Too large a  $k$  will lead to an unduly large complexity. Too small a  $k$  will lead to loss of information (and subsequently excessively uncertain predictions from training), and might make assertion 2 of our prior specification inappropriate. For a given  $k$ , if the inducing points are not carefully chosen, the coverage of the domain will not be adapted to changes in the intensity function and as a result, the predictive variance in certain parts of the domain might considerably differ from the

posterior variance we would have obtained, had we chosen inducing points in those parts of the domain.

Intuitively, a good algorithm to find inducing points should leverage prior knowledge about the smoothness, periodicity, amplitude and length scale(s) of the intensity function to optimize for the quality of (post-training) predictions while minimising the number of inducing points.

We use as utility function for the choice of inducing points:

$$\mathcal{U}(\mathcal{D}') = \mathbb{E}_\theta(\text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}'}^*(\theta) \Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\theta) \Sigma_{\mathcal{D}\mathcal{D}'}^{*T}(\theta))) \quad (13)$$

where  $\theta$  is the vector of hyper-parameters of the covariance kernel  $\gamma^*$ , and the expectation is taken with respect to the prior distribution over  $\theta$ . In other words, the utility of a set of inducing points is the expected total reduction of the (predictive) variances of  $\log \lambda(s_1), \dots, \log \lambda(s_n)$  resulting from knowing  $\log \lambda(s'_1), \dots, \log \lambda(s'_k)$ .

In practice, the expectation in Equation (13) might not be available analytically. We can however use the Monte Carlo estimate:

$$\tilde{\mathcal{U}}(\mathcal{D}') = \frac{1}{N} \sum_{i=1}^N \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}'}^*(\tilde{\theta}_i) \Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\tilde{\theta}_i) \Sigma_{\mathcal{D}\mathcal{D}'}^{*T}(\tilde{\theta}_i)). \quad (14)$$

The algorithm proceeds as follows. We sample  $(\tilde{\theta}_i)_{i=1}^N$  from the prior. Initially we set  $k = 0$ ,  $\mathcal{D}' = \emptyset$  and  $u_0 = 0$ . We increment  $k$  by one, and consider adding an inducing point. We then find the point  $s'_k$  that maximises  $\tilde{\mathcal{U}}(\mathcal{D}' \cup \{s\})$

$$s'_k := \underset{s \in \mathcal{S}}{\text{argmax}} \tilde{\mathcal{U}}(\mathcal{D}' \cup \{s\}) \quad (15)$$

using Bayesian optimisation (Mockus, 2013). We compute the utility of having  $k$  inducing points as

$$u_k = \tilde{\mathcal{U}}(\mathcal{D}' \cup \{s'_k\}),$$

we update  $\mathcal{D}' = \mathcal{D}' \cup \{s'_k\}$  and stop when

$$\frac{u_k - u_{k-1}}{u_k} < \alpha,$$

where  $0 < \alpha \ll 1$  is a convergence threshold.

### Proposition

(a) For any  $\mathcal{D}$ ,  $\alpha$ ,  $N$  and  $p_\theta$  Algorithm 1 stops in finite time and the sequence  $(u_k)_{k \in \mathbb{N}}$  converges at least linearly with rate  $1 - \frac{1}{\#\mathcal{D}}$ .

(b) Moreover, the maximum utility  $u_f(\alpha)$  returned by Algorithm 1 converges to the average total unconditional variance  $w_\infty := \frac{1}{N} \sum_{i=1}^N \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}}^*(\tilde{\theta}_i))$  as  $\alpha$  goes to 0.

The idea behind the proof of this proposition is that the sequence of maximum utilities  $u_k$  is positive, increas-



**Algorithm 1** Selection of inducing points

---

**Inputs:**  $0 < \alpha \ll 1, N, p_\theta$   
**Output:**  $u_f, \mathcal{D}'$   
 $k = 0, u_0 = 0, \mathcal{D}' = \emptyset, e = 1$ ;  
 Sample  $(\tilde{\theta}_i)_{i=1}^N$  from  $p(\theta)$ ;  
**while**  $e > \alpha$  **do**  
      $k = k + 1$ ;  
      $s'_k = \underset{s \in \mathcal{S}}{\operatorname{argmax}} \tilde{\mathcal{U}}(\mathcal{D}' \cup \{s\})$ ;  
      $u_k = \tilde{\mathcal{U}}(\mathcal{D}' \cup \{s'_k\})$ ;  
      $\mathcal{D}' = \mathcal{D}' \cup \{s'_k\}$ ;  
      $e = \frac{u_k - u_{k-1}}{u_k}$ ;  
**end while**

---

ing<sup>2</sup>, and upper-bounded by the total unconditional variance  $w_\infty$ <sup>3</sup>. Hence, the sequence  $u_k$  converges to a strictly positive limit, which implies that the stopping condition of the while loop will be met in finite time regardless of  $\mathcal{D}$ ,  $\alpha$ ,  $N$  and  $p_\theta$ . Finally, we construct a sequence  $w_k$  upper-bounded by the sequence  $u_k$  and that converges linearly to the average total unconditional variance  $w_\infty$  with rate  $1 - \frac{1}{\#\mathcal{D}}$ . As the sequence  $u_k$  converges and is itself upper-bounded by  $w_\infty$ , its limit is  $w_\infty$  as well, and it converges at least as fast as  $w_k$ . (See appendix for the full proof)

Our algorithm is particularly suitable to Poisson Point Processes as it prioritises sampling inducing points in parts of the domain where the data are denser. This corresponds to regions where the intensity function will be higher, thus where the local random counts of the underlying PPP will vary more<sup>4</sup> and subsequently where the posterior variance of the intensity is expected to be higher. Moreover, it leverages prior smoothness assumptions on the intensity function to limit the number of inducing points and to appropriately and sequentially improve coverage of the domain.

Algorithm 1 is illustrated on a variety of real life and synthetic data sets in section 5.

## 4. INFERENCE

We use a Squared Exponential kernel for  $\gamma^*$  and *Scaled Sigmoid Gaussian* priors for the kernel hyper-parameters; that is  $\theta_i = \frac{\theta_{\max}}{1 + \exp(-x_i)}$  where  $x_i$  are i.i.d standard Normal. The problem-specific scales,  $\theta_{\max}$ , restrict the supports of those distributions using prior knowledge to avoid unlikely extreme values and to improve conditioning.

We use a Block Gibbs Sampler (Geman & Geman, 1984)

<sup>2</sup>Intuitively, conditioning on a new point increases the reduction of variance from the unconditional variance.

<sup>3</sup>The variance cannot be reduced by more than the total unconditional variance.

<sup>4</sup>The variance of the Poisson distribution is its mean.

to sample from the posterior. We sample the hyper-parameters using the Metropolis-Hastings (Hastings, 1970) algorithm taking as proposal distribution the prior of the variable of interest. We sample the log-intensities at the inducing points using Elliptical Slice Sampling (Murray et al., 2010) with the pdf in Equation (12).

### Prediction from training

To predict the posterior mean at the data points we note from the law of total expectation that

$$\begin{aligned} \forall s_i \in \mathcal{D}, \mathbb{E}(\log \lambda(s_i) | \mathcal{D}) \\ = \mathbb{E} \left( \mathbb{E} \left( \log \lambda(s_i) | \{\log \lambda^*(s'_j)\}_{j=1}^k, \mathcal{D} \right) | \mathcal{D} \right). \end{aligned} \quad (16)$$

Also, we note from Equations (1) and (5) that the dependency of the posterior of  $\log \lambda(s_i)$  conditional on  $\{\log \lambda^*(s'_j)\}_{j=1}^k$  is of the form

$$\exp(\log \lambda(s_i)) \times \mathcal{N}(\log \lambda(s_i) | m(s_i), \gamma(s_i, s_i)),$$

where we recall that  $m(s_i)$  is the  $i$ -th element of the vector  $M$  and  $\gamma(s_i, s_i)$  is the  $i$ -th diagonal element of the matrix  $\Sigma_{\mathcal{D}\mathcal{D}}$ . Hence, the posterior distribution of  $\log \lambda(s_i)$  conditional on  $\{\log \lambda^*(s'_j)\}_{j=1}^k$  is Gaussian with mean

$$\mathbb{E} \left( \log \lambda(s_i) | \{\log \lambda^*(s'_j)\}_{j=1}^k, \mathcal{D} \right) = M[i] + \Sigma_{\mathcal{D}\mathcal{D}}[i, i] \quad (17)$$

and variance

$$\text{Var} \left( \log \lambda(s_i) | \{\log \lambda^*(s'_j)\}_{j=1}^k, \mathcal{D} \right) = \Sigma_{\mathcal{D}\mathcal{D}}[i, i]. \quad (18)$$

Finally, it follows from Equation (16) that  $\mathbb{E}(\log \lambda(s_i) | \mathcal{D})$  is obtained by averaging out  $M[i] + \Sigma_{\mathcal{D}\mathcal{D}}[i, i]$  over MCMC samples after burn-in.

Similarly, the law of total variance implies that

$$\begin{aligned} \text{Var}(\log \lambda(s_i) | \mathcal{D}) \\ = \mathbb{E} \left( \text{Var} \left( \log \lambda(s_i) | \{\log \lambda^*(s'_j)\}_{j=1}^k, \mathcal{D} \right) | \mathcal{D} \right) \\ + \text{Var} \left( \mathbb{E} \left( \log \lambda(s_i) | \{\log \lambda^*(s'_j)\}_{j=1}^k, \mathcal{D} \right) | \mathcal{D} \right). \end{aligned} \quad (19)$$

Hence, it follows from Equations (17) and (18) that the posterior variance at a data point  $s_i$  is obtained by summing up the sample mean of  $\Sigma_{\mathcal{D}\mathcal{D}}[i, i]$  with the sample variance of  $M[i] + \Sigma_{\mathcal{D}\mathcal{D}}[i, i]$ , where sample mean and sample variance are taken over MCMC samples after burn-in.

## 5. EXPERIMENTS

We selected four data sets to illustrate the performance of our model. We restricted ourselves to one synthetic data set for brevity. We chose the most challenging of the synthetic intensity functions of (Adams et al., 2009) and (Rao & Teh, 2011),  $\lambda(t) = 2 \exp(-\frac{t}{15}) + \exp(-(\frac{t-25}{10})^2)$ ,

Table 1. Maximum output (resp. input) scale  $h_{\max}$  (resp.  $l_{\max}$ ) used for each data set to select inducing points.

	SYNTHETIC	COAL MINE	BRAMBLE	TWITTER
$h_{\max}$	10.0	10.0	10.0	10.0
$l_{\max}$	25.0	50.0	0.25	5.0

Table 2. Number of inducing points produced by Algorithm 1 required to achieve some critical normalised utility values on the 4 data sets.

	$k$			
$\frac{u_k}{u_\infty}$	SYNTHETIC	COAL MINE	BRAMBLE	TWITTER
0.75	2	2	8	3
0.90	3	4	17	5
0.95	4	5	28	8

to thoroughly compare our model with competing methods. We also ran our model on a standard 1 dimensional real-life data set (the coal mine disasters dataset used in (Jarrett, 1979); 191 points) and a standard real-life 2 dimensional data (spatial location of bramble canes (Diggle, 1983); 823 points). Finally we ran our model on a real-life data set large enough to cause problems to competing models. This data set consists of the UTC timestamps (expressed in hours in the day) of Twitter updates in English published in the (Twitter Sample Stream, 2014) on September 1st 2014 (188544 points).

### 5.1. Inducing points selection

Figure 1 illustrates convergence of the selection of inducing points on the 4 data sets. We ran the algorithm 10 times with  $N = 20$ , and plotted the average normalised utility  $\frac{u_k}{u_\infty} \pm 1$  std as a function of the number of inducing points. Table 1 contains the maximum hyper-parameters that were used for each data set. Table 2 contains the number of inducing points required to achieve some critical normalised utility values for each of the 4 data sets. We note that just 8 inducing points were required to achieve a 95% utility for the Twitter data set (188544 points). In regards to the positions of sampled inducing points, we note from Figures 2 and 3 that when the intensity function was bimodal, the first inducing point was sampled around the argument of the highest mode, and the second inducing point was sampled around the argument of the second highest mode. More generally, the algorithm sampled inducing points where the latent intensity function varies the most, as expected.

### 5.2. Intensity function

In each experiment we generated 5000 samples after burn-in (1000 samples). For each data set we used the set of

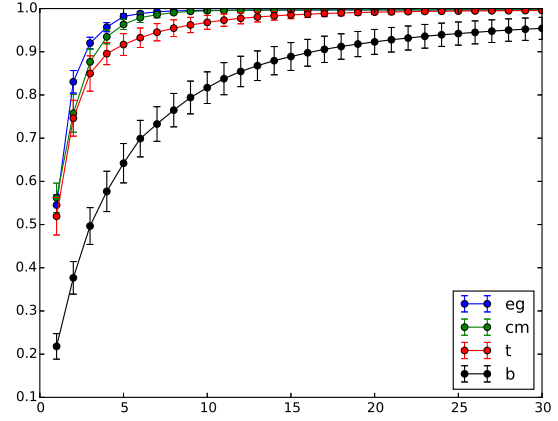


Figure 1. Average normalised utility  $\frac{u_k}{u_\infty}$  of choosing  $k$  inducing points using Algorithm 1  $\pm 1$  standard deviation as a function of  $k$  on the synthetic data set (eg), the coal mine data set (cm), the Twitter data set (t) and the bramble canes data set (b). The average was taken over 10 runs.

inducing points that yielded a 95% normalized utility. The exact numbers are detailed in Table 2.

We ran a Monte Carlo simulation for the stochastic processes considered herein and found that the Legendre polynomial order  $p = 10$  was sufficient to yield a Quadrature estimate for the standard deviation of the integral less than 1% away from the Monte Carlo estimate (using the trapezoidal rule), and a Quadrature estimate for the mean of the integral less than a standard error away from the Monte Carlo average. We took a more conservative stand and used  $p = 20$ .

### Inference on synthetic data

We generated a draw from a Poisson point process with the intensity function  $\lambda(t) = 2 \exp(-\frac{t}{15}) + \exp(-(\frac{t-25}{10})^2)$  of (Adams et al., 2009) and (Rao & Teh, 2011). The draw consisted of 41 points (blue sticks in Figure 2). We compared our model to (Adams et al., 2009) (SGCP) and (Rao & Teh, 2011) (RMP). We ran the RMP model with the renewal parameter  $\gamma$  set to 1 (RMP 1), which corresponds to an exponential renewal distribution or equivalently an inhomogeneous Poisson process. We also ran the RMP model with a uniform prior on  $[1, 5]$  over the renewal parameter  $\gamma$  (RMP full). Figure 2 illustrates the posterior mean intensity function under each model. Finally we ran the Dirichlet Process Mixture of Beta model of (Kottas, 2006) (DPMB). As detailed in Table 3, our model outperformed that of (Adams et al., 2009), (Rao & Teh, 2011) and (Kottas, 2006) in terms of accuracy and speed.

### Inference on real-life data

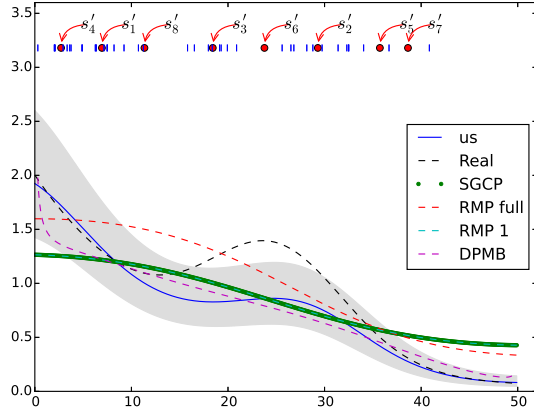


Figure 2. Inference on a draw (blue sticks) from a Poisson point process with intensity  $\lambda(t) = 2\exp(-\frac{t}{15}) + \exp(-(\frac{t-25}{10})^2)$  (black line). The red dots are the inducing points generated by our algorithm, labelled in the order they were selected. The solid blue line and the grey shaded area are the posterior mean  $\pm 1$  posterior standard deviation under our model. SGCP is the posterior mean under (Adams et al., 2009). RMP full and RMP 1 are the posterior mean intensities under (Rao & Teh, 2011) with  $\gamma$  inferred and set to 1 respectively. DPMB is the Dirichlet Process mixture of Beta (Kottas, 2006)

Figure 3 shows the posterior mean intensity functions of the coal mine data set, the Twitter data set and the bramble canes data set under our model.

**Scalability:** We note that it took only 240s on average to generate 1000 samples on the Twitter data set (188544 points). As a comparison, this is the amount of time that would be required to generate as many samples on a data set that has 50 points (resp. 100 points) under the models of (Adams et al., 2009) (resp. (Rao & Teh, 2011)). More importantly, it was not possible to run either of those two

Table 3. Some statistics on the MCMC runs of Figure 2. RMSE and MAE denote the Root Mean Square Error and the Mean Absolute Error, expressed as a proportion of the average of the true intensity function over the domain. LP denotes the log mean predictive probability on 10 held out PPP draws from the true intensity  $\pm 1$  std. t(s) is the average time in seconds it took to generate 1000 samples  $\pm 1$  std and ESS denotes the average effective sample size (Gelman et al., 2013) per 1000 samples.

	MAE	RMSE	LP	T (s)	ESS
SGCP	0.31	0.37	-45.07 $\pm$ 1.64	257.72 $\pm$ 16.29	6
RMP 1	0.32	0.38	-45.24 $\pm$ 1.41	110.19 $\pm$ 7.37	23
RMP FULL	0.25	0.31	-43.51 $\pm$ 2.15	139.64 $\pm$ 5.24	6
DPMB	0.23	0.32	-42.95 $\pm$ 3.58	23.27 $\pm$ 0.94	47
Us	0.19	0.27	-42.84 $\pm$ 3.07	4.35 $\pm$ 0.12	38

competing models on the twitter data set. Doing so would require computing  $17 \times 10^{10}$  covariance coefficients to evaluate a single auto-covariance matrix of the log-intensity at the data points, which a typical personal computer cannot handle.

## 6. DISCUSSION

### Scalability of the selection of inducing points

The computational bottleneck of the selection of inducing points is in the evaluation of

$$\text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}'}^*(\tilde{\theta}_i)\Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\tilde{\theta}_i)\Sigma_{\mathcal{D}\mathcal{D}'}^*(\tilde{\theta}_i)).$$

Hence, the complexity and the memory requirement of the selection of inducing points are both linear in the number of data points  $n := \#\mathcal{D}$ .

The number of inducing points generated by our algorithm does not increase with the size of the data, but rather as a function of the size of the domain and the resolution implied by the prior over the hyper-parameters.

### Comparison with competing models

We note that the computational bottleneck of our MCMC inference is in the evaluation of

$$\text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}}) = \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}}^*) - \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}'}^*\Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}\Sigma_{\mathcal{D}\mathcal{D}'}^*).$$

Hence, inferring the intensity function under our model scales computationally in  $\mathcal{O}(nk^2)$  and has a memory requirement  $\mathcal{O}(nk)$ , where the number of inducing points  $k$  is negligible. This is considerably better than alternative methods using Gaussian Processes (Adams et al., 2009; Rao & Teh, 2011) whose complexities are cubic in the number of data points and whose memory requirement is squared in the number of data points. Moreover, the superior accuracy of our model compared to (Adams et al., 2009) and (Rao & Teh, 2011) is due to our use of the exponential transformation rather than the scaled sigmoid one. In effect, unlike the inverse scaled sigmoid function that tends to amplify variations, the logarithm tends to smooth out variations. Hence, when the true intensity is uneven, the log-intensity is more likely to resemble a draw from a stationary GP than the inverse scaled sigmoid of the true intensity function, and subsequently a stationary GP prior in the inverse domain is more suitable to the exponential transformation than to the scaled sigmoid transformation.

Our model is also more suitable than that of (Cunningham et al., 2008a) when confidence bounds are needed for the intensity function, or when the input space is of dimension higher than 1. The model is a useful alternative to that of (Kottas, 2006), whose complexity is also linear. In effect, Gaussian Processes (GP) are more flexible than a Dirichlet Process (DP) mixture of Beta

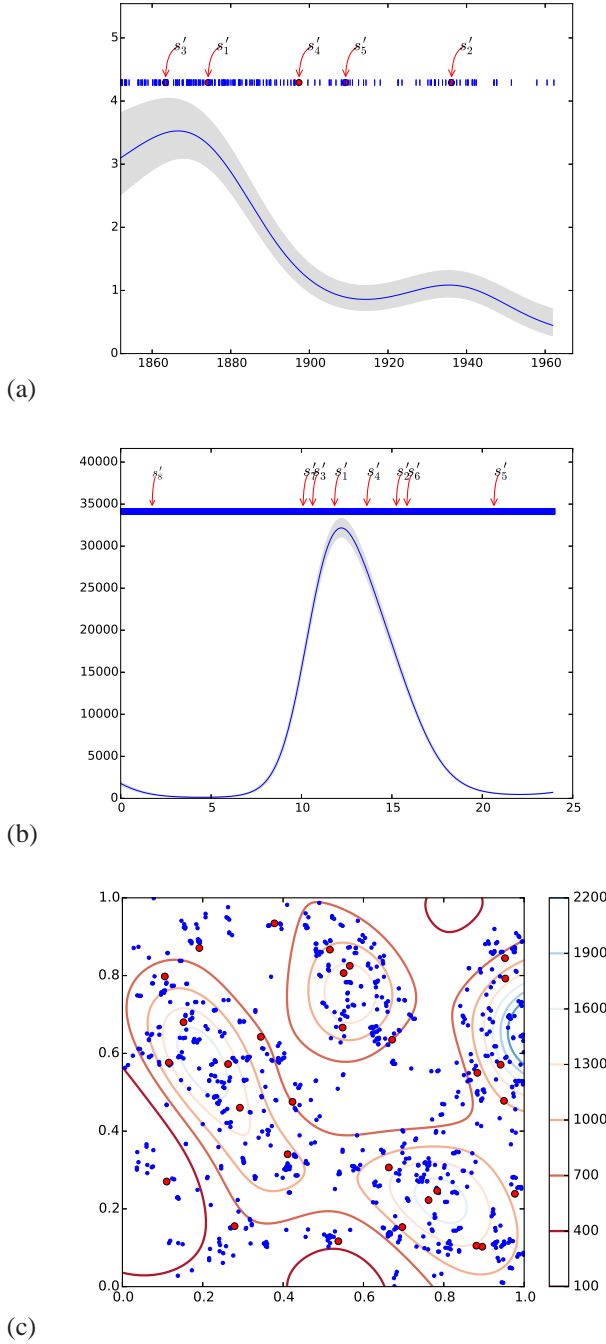


Figure 3. Inference on the intensity functions of the coal mine data set (top), the twitter data set (middle), and the bramble canes data set (bottom). Blue dots are data points, red dots are inducing points (labelled in the upper panels in the order they were selected), the grey area is the 1 standard deviation confidence band.

distributions. This is the result of the large number of known covariance kernels available in the literature and the state-of-the-art understanding of how well a given kernel

can approximate an arbitrary function (Micchelli et al., 2006; Pillai et al., 2007). Moreover, unlike a Dirichlet Process mixture of Beta distributions, Gaussian Processes allow directly expressing practical prior features such as smoothness, amplitude, length scale(s) (memory), and periodicity.

As our model relies on the Gauss-Legendre quadrature, we would not recommend it for applications with a large input space dimension. However, most interesting point process applications involve modelling temporal, spatial or spatio-temporal events, for which our model scales considerably better with the data size than competing approaches. In effect, the models proposed by (Kottas, 2006; Cunningham et al., 2008a;b; Rao & Teh, 2011) are all specific to unidimensional input data, whereas the model introduced by (Kottas & Sanso, 2007) is specific to spatial data. As for the model of (Adams et al., 2009), it scales very poorly with the input space dimension for its complexity is cubic in the sum of the number of data points and the number of latent thinning points, and the number of thinning points grows exponentially with the input space dimension<sup>5</sup>.

### Extension of our model

Although the covariance kernel  $\gamma^*$  was assumed stationary, no result in this paper relied on that assumption. We solely needed to evaluate covariance matrices under  $\gamma^*$ . Hence, the proposed model and algorithm can also be used to account for known non-stationarities. More generally, the model presented in this paper can serve as foundation to make inference on the stochastic dependency between multiple point processes when the intensities are assumed to be driven by known exogenous factors, hidden common factor, and latent idiosyncratic factors.

## 7. SUMMARY

In this paper we propose a novel exact non-parametric model to make inference on Poisson Point Processes using Gaussian Processes. We derive a robust MCMC scheme to sample from the posterior intensity function. Our model outperforms competing benchmarks in terms of speed and accuracy as well as in the decorrelation of MCMC samples. A critical advantage of our approach is that it has a numerical complexity and a memory requirement *linear* in the data size  $n$  ( $\mathcal{O}(nk^2)$ , and  $\mathcal{O}(nk)$  respectively, with  $k \ll n$ ). Competing models using Gaussian Processes have a cubic numerical complexity and squared memory requirement. We show that our model readily handles data sizes

<sup>5</sup>The expected number of thinning points grows proportionally with the volume of the domain, which is exponential in the dimension of the input space when the domain is a hypercube with a given edge length.



not yet considered in the literature.

## Acknowledgments

Yves-Laurent Kom Samo is supported by the Oxford-Man Institute of Quantitative Finance.

## Appendix

### A. There exists a Cox process with an a.s. $C^\infty$ intensity coinciding with any finite dimensional prior.

In this section we prove the proposition below.

**Proposition .1** *Let  $\mathbb{Q}$  be an  $(n + 1)$  dimensional continuous probability distribution whose density has support  $\bigotimes_{i=1}^{n+1} ]0, +\infty[$ , and let  $x_1, \dots, x_n$  be  $n$  points on a compact domain  $\mathcal{S} \subset \mathbb{R}^d$ . There exists an almost surely non-negative and  $C^\infty$  stochastic process  $\lambda$  on  $\mathcal{S}$  such that*

$$(\lambda(x_1), \dots, \lambda(x_n), \int_{\mathcal{S}} \lambda(x) dx) \sim \mathbb{Q}.$$

**Proof** Let

$$(y_1, \dots, y_n, I) \sim \mathbb{Q}$$

and

$$(y_1(\omega), \dots, y_n(\omega), I(\omega))$$

a random draw. Let us denote  $x^j, j \leq d$  the  $j$ -th coordinate of  $x \in \mathbb{R}^d$ . We consider the family of functions parametrized by  $\alpha \in \mathbb{R}$ :

$$\begin{aligned} f(\omega, x, \alpha) = \exp \left( \alpha \sum_{j=1}^d \prod_{l=1}^n (x^j - x_l^j)^2 \right) \quad (20) \\ \times \sum_{l=1}^n y_l(\omega) \frac{1}{d} \sum_{j=1}^d \prod_{k \neq l} \left( \frac{x^j - x_k^j}{x_l^j - x_k^j} \right)^2. \end{aligned}$$

We note that  $\forall \alpha, x_i, f(\omega, x_i, \alpha) = y_i(\omega)$ . Let us define the polynomial

$$P(x) = \sum_{l=1}^n y_l(\omega) \frac{1}{d} \sum_{j=1}^d \prod_{k \neq l} \left( \frac{x^j - x_k^j}{x_l^j - x_k^j} \right)^2.$$

As  $P$  is continuous, it is bounded on the compact  $\mathcal{S}$ , and reaches its bounds. Thus we have

$$\exists m_p, M_p \geq 0, \text{ s.t. } \forall x \in \mathcal{S}, 0 \leq m_p \leq P(x) \leq M_p.$$

Similarly, if we define

$$R(\alpha, x) = \exp \left( \alpha \sum_{j=1}^d \prod_{l=1}^n (x^j - x_l^j)^2 \right) = R(1, x)^\alpha,$$

it follows that

$$\exists m_q, M_q > 1, \text{ s.t. } \forall x \in \mathcal{S}, 1 < m_q \leq R(1, x) \leq M_q.$$

Hence,

$$m_p m_q^\alpha \mu(\mathcal{S}) \leq \int_{\mathcal{S}} f(\omega, x, \alpha) dx \leq M_p M_q^\alpha \mu(\mathcal{S}). \quad (21)$$

Moreover, we note that  $\alpha \rightarrow \int_{\mathcal{S}} f(\omega, x, \alpha) dx$  is continuous on  $\mathbb{R}$  as its restriction to any bounded interval is continuous (by dominated convergence theorem). Furthermore, given that  $m_q, M_q > 1$ , it follows from Equation (21) that

$$\lim_{\alpha \rightarrow +\infty} \int_{\mathcal{S}} f(\omega, x, \alpha) dx = +\infty$$

and

$$\lim_{\alpha \rightarrow -\infty} \int_{\mathcal{S}} f(\omega, x, \alpha) dx = 0.$$

Hence, by intermediate value theorem,

$$\forall I(\omega) > 0, \exists \alpha^*(\omega) \text{ s.t. } I(\omega) = \int_{\mathcal{S}} f(\omega, x, \alpha^*(\omega)) dx.$$

Finally, let us define the stochastic process  $\lambda$  on  $\mathcal{S}$  as

$$\omega \rightarrow \lambda(\omega, x) := f(\omega, x, \alpha^*(\omega)).$$

To summarise,

$$\forall x_i, \lambda(\omega, x_i) := f(\omega, x_i, \alpha^*(\omega)) = y_i(\omega),$$

$$I(\omega) = \int_{\mathcal{S}} \lambda(\omega, x) dx,$$

and

$$(y_1, \dots, y_n, I) \sim \mathbb{Q} :$$

this implies  $(\lambda(x_1), \dots, \lambda(x_n), \int_{\mathcal{S}} \lambda(x) dx) \sim \mathbb{Q}$ . Finally,

$$\forall x \in \mathcal{S}, \lambda(\omega, x) \geq 0, \text{ and } \forall \omega, x \rightarrow \lambda(\omega, x) \text{ is } C^\infty,$$

which concludes our proof.

### B. Proof of convergence of Algorithm 1

The idea behind the proof is to show that the sequence of maximum utility

$$u_k = \max_{s \in \mathcal{S}} \tilde{\mathcal{U}}(\{s'_1, \dots, s'_{k-1}\} \cup \{s\})$$

is positive, increasing and upper-bounded and thus converges to a strictly positive limit. This would then imply that

$$\frac{u_{k+1} - u_k}{u_k} \xrightarrow[k \rightarrow \infty]{} 0$$

and subsequently that

$$\forall 0 < \alpha < 1, \exists k_{\text{lim}} \in \mathbb{N} \text{ s.t. } \forall k > k_{\text{lim}}, \frac{u_{k+1} - u_k}{u_k} < \alpha$$

or in other words Algorithm 1 always stops in finite time.

To show that  $\forall k > 0, u_k > 0$ , we note that  $\Sigma_{\mathcal{D}'\mathcal{D}'}^*(\tilde{\theta}_i)$  is a covariance matrix and as such it is positive definite. It follows that  $\Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\tilde{\theta}_i)$  is also positive definite. We further note that the  $j$ -th diagonal term of  $\Sigma_{\mathcal{D}'\mathcal{D}'}^*(\tilde{\theta}_i)\Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\tilde{\theta}_i)\Sigma_{\mathcal{D}'\mathcal{D}'}^*(\tilde{\theta}_i)$  can be written as  $x_j^T \Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\tilde{\theta}_i)x_j$  where  $x_j$  is the  $j$ -th column of  $\Sigma_{\mathcal{D}'\mathcal{D}'}^*(\tilde{\theta}_i)$ . Hence, by virtue of the positive definiteness of  $\Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\tilde{\theta}_i)$ , the diagonal terms of  $\Sigma_{\mathcal{D}'\mathcal{D}'}^*(\tilde{\theta}_i)\Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\tilde{\theta}_i)\Sigma_{\mathcal{D}'\mathcal{D}'}^*(\tilde{\theta}_i)$  are all positive, which proves that the utility function  $\tilde{\mathcal{U}}$  is positive, and subsequently that  $\forall k > 0, u_k > 0$ .

To show that  $(u_k)_{k \in \mathbb{N}^*}$  is upper-bounded, we note that the matrix

$$C_{i\mathcal{D}'} = \Sigma_{\mathcal{D}\mathcal{D}'}^*(\tilde{\theta}_i) - \Sigma_{\mathcal{D}\mathcal{D}'}^*(\tilde{\theta}_i)\Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\tilde{\theta}_i)\Sigma_{\mathcal{D}'\mathcal{D}'}^*(\tilde{\theta}_i)$$

where the notation is as per the rest of the paper, is an auto-covariance matrix, and as such has positive diagonal elements. Hence,

$$\text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}}^*(\tilde{\theta}_i)) \geq \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}'}^*(\tilde{\theta}_i)\Sigma_{\mathcal{D}'\mathcal{D}'}^{*-1}(\tilde{\theta}_i)\Sigma_{\mathcal{D}'\mathcal{D}'}^*(\tilde{\theta}_i))$$

and finally

$$\forall k \in \mathbb{N}^*, u_k \leq \frac{1}{N} \sum_{i=1}^N \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}}^*(\tilde{\theta}_i)).$$

Moreover, we note that showing that  $(u_k)_{k \in \mathbb{N}^*}$  is increasing is equivalent to showing that  $(v_k)_{k \in \mathbb{N}^*}$  with

$$v_k = \min_{s \in \mathcal{S}} \frac{1}{N} \sum_{i=1}^N \text{Tr}(C_{i\{s'_1, \dots, s'_{k-1}\} \cup \{s\}})$$

is decreasing. We recall that  $C_{i\{s'_1, \dots, s'_{k-1}\} \cup \{s\}}$  is the covariance matrix of the values of the stationary Gaussian Process of our model at the data points, conditioned on its values at  $\{s'_1, \dots, s'_{k-1}\} \cup \{s\}$ .

It follows from the law of iterated expectations that  $C_{i\{s'_1, \dots, s'_{k-1}\} \cup \{s\}}$  could also be seen as the covariance matrix of the values of a conditional Gaussian Process at the data points,<sup>6</sup> conditioned on its value at  $s$ . Hence,

$$C_{i\{s'_1, \dots, s'_{k-1}\} \cup \{s\}} = C_{i\{s'_1, \dots, s'_{k-1}\}} - \frac{1}{\hat{\Sigma}_{ss}(\tilde{\theta}_i)} \hat{\Sigma}_{\mathcal{D}\{s\}}(\tilde{\theta}_i) \hat{\Sigma}_{\mathcal{D}\{s\}}^T(\tilde{\theta}_i)$$

where  $\hat{\Sigma}_{XY}$  denotes the covariance matrix between the values of the conditional GP at points in  $X$  and at points in

<sup>6</sup>The conditional GP is defined as the stationary Gaussian Process in our model is conditioned on its values at the points  $\{s'_1, \dots, s'_{k-1}\}$

$Y$ . In particular,  $\hat{\Sigma}_{ss}(\tilde{\theta}_i)$  is a positive scalar. What's more the diagonal elements of  $\hat{\Sigma}_{\mathcal{D}\{s\}}(\tilde{\theta}_i)\hat{\Sigma}_{\mathcal{D}\{s\}}^T(\tilde{\theta}_i)$  are all non-negative. Hence,

$$\forall s \in \mathcal{S}, \text{Tr}(C_{i\{s'_1, \dots, s'_{k-1}\} \cup \{s\}}) \leq \text{Tr}(C_{i\{s'_1, \dots, s'_{k-1}\}})$$

and averaging over the set of hyper-parameters  $\theta_i$  and taking the min we get

$$\forall k \geq 2, v_k \leq v_{k-1}$$

which concludes the proof.

### C. Proof of the rate of convergence of Algorithm 1 and that $u_f$ in Algorithm 1 converges to $\frac{1}{N} \sum_{i=1}^N \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}}^*(\tilde{\theta}_i))$ as $\alpha$ goes to 0

The key idea of this proof is to note as previously shown that no set of inducing points has a utility greater than  $w_\infty := \frac{1}{N} \sum_{i=1}^N \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}}^*(\tilde{\theta}_i))$ , but that any set of inducing points that includes  $\mathcal{D}$  has a utility equal to  $w_\infty$ .

Let  $\{s'_1, \dots, s'_k\}$  be points selected after  $k$  iterations of Algorithm 1, and let us denote by  $\{u_1, \dots, u_k\}$  the maximum utilities after the corresponding iterations as usual. Let us denote by

$$\tilde{s}_k = \underset{s \in \mathcal{D}}{\text{argmax}} \tilde{\mathcal{U}}(\{s'_1, \dots, s'_{k-1}\} \cup \{s\})$$

the best candidate *in the data set* to be the  $k$ -th inducing point after  $k-1$  iterations of our algorithm. As previously mentioned,  $\{s'_1, \dots, s'_{k-1}\} \cup \mathcal{D}$  is a set of inducing points with perfect utility. Therefore, if we select the data points as inducing points after  $\{s'_1, \dots, s'_{k-1}\}$ , their contribution to the overall utility will be  $w_\infty - u_{k-1}$ . If we further constrain our choice of  $\mathcal{D}$  as additional inducing points to start with  $\tilde{s}_k$  then the incremental utility of choosing  $\tilde{s}_k$  will be at least  $\frac{w_\infty - u_{k-1}}{n}$ , where  $n$  is the data size as usual. This is because  $\tilde{s}_k$  is the best choice for the  $k$ -th inducing point in  $\mathcal{D}$  after having picked  $\{s'_1, \dots, s'_{k-1}\}$  and because the incremental utility of choosing an inducing point is higher earlier (when little is known about the GP) than later (when more is known about the GP). What's more, by definition, the incremental utility of choosing  $s'_k$  after  $\{s'_1, \dots, s'_{k-1}\}$  is higher than that of choosing  $\tilde{s}_k$  after  $\{s'_1, \dots, s'_{k-1}\}$ . Hence,

$$u_k - u_{k-1} \geq \frac{w_\infty - u_{k-1}}{n}.$$

Let us denote by  $w_k$  the sequence satisfying

$$w_0 = u_0, \forall k \in \mathbb{N}^* w_k - w_{k-1} = \frac{w_\infty - w_{k-1}}{n}.$$

It can be shown (by induction on  $k$ ) that

$$\forall k \in \mathbb{N}^* w_k \leq u_k.$$

Moreover, we note that

$$w_k - w_\infty = (1 - \frac{1}{n})(w_{k-1} - w_\infty).$$

Hence

$$w_k = w_\infty + (1 - \frac{1}{n})^k (w_0 - w_\infty),$$

which proves that the sequence  $w_k$  converges linearly to  $w_\infty$  with rate  $1 - \frac{1}{n}$ .

On one hand, we have shown that the sequence  $u_k$  converges and is upper-bounded by  $w_\infty$ , hence its limit is smaller than  $w_\infty$ :

$$u_\infty := \lim_{k \rightarrow \infty} u_k \leq w_\infty.$$

On the other hand, we have shown that  $\forall k \in \mathbb{N}^* w_k \leq u_k$  which implies

$$w_\infty \leq u_\infty.$$

Hence,

$$w_\infty = u_\infty = \frac{1}{N} \sum_{i=1}^N \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}}^*(\tilde{\theta}_i)).$$

As  $w_k$  is upper-bounded by  $u_k$  and both sequences converge to the same limit,  $u_k$ , and subsequently Algorithm 1, converge at least as fast as  $w_k$ .

In regards to the second statement of our proposition, we have that

$$\lim_{\alpha \rightarrow 0} u_f(\alpha) = \lim_{k \rightarrow \infty} u_k = \frac{1}{N} \sum_{i=1}^N \text{Tr}(\Sigma_{\mathcal{D}\mathcal{D}}^*(\tilde{\theta}_i)).$$

## References

- Adams, R.P., Murray, I., and MacKay, D.J.C. Tractable nonparametric bayesian inference in Poisson processes with gaussian process intensities. pp. 9–16, 2009.
- Basu, S. and Dassios, A. (2002) A Cox process with log-normal intensity. *Insurance: mathematics and economics*, 31 (2). pp. 297–302. ISSN 0167-6687
- Cox, D.R. Some Statistical Methods Connected with Series of Events. *Journal of the Royal Statistical Society*, 17: 129–164, 1955.
- Cox, D.R., Isham, V. (eds.). *Point Processes*. Chapman Hall/CRC, 1980.
- Cunningham, J.P., Shenoy, K.V., and Sahani, M. Fast Gaussian Process Methods for Point Process Intensity Estimation. Appearing in Proceedings of the 25 th International Conference on Machine Learning, Helsinki, Finland, 2008.
- Cunningham, J.P., Yu, B., Shenoy, K.V., and Sahani, M. Inferring neural firing rates from spike trains using Gaussian Processes. *Advances in Neural Information Processing Systems* 20 (pp. 329336).
- Daley, D.J. and Vere-Jones, D. *An Introduction to the Theory of Point Processes*. Springer-Verlag, 2008.
- Diggle, P.J. *Statistical Analysis of Spatial Point Patterns*. Academic Press.
- Diggle, P.J. *A kernel method for smoothing point process data*. *Applied Statistics*, 34:138–147, 1985.
- Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., and Rubin, D.B. (eds.). *Bayesian Data Analysis Thrid Edition*. CRC Press, 2013.
- Geman, S. and Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- Gregory, P. C., Lored, T. J. A new method for the detection of a periodic signal of unknown shape and period. *The Astrophysical Journal*, The Astrophysical Journal, 398, 146168, 1992.
- Gunter, T., Lloyd, C., Osborne, M.A., Roberts, S.J. Efficient Bayesian Nonparametric Modelling of Structured Point Processes. *Uncertainty in Artificial Intelligence (UAI)*, 2014.
- Hastings, W.K. Monte Carlo sampling methods using markov chains and their applications. *Biometrika*, 24: 97–109, 1970.

- Heikkinen, J., Arjas, E. Modeling a Poisson forest in variable elevations: a nonparametric Bayesian approach. *Biometrics*, 55, 738745, 1999.
- Hildebrand, F. B, *Introduction to Numerical Analysis: Second Edition*, Chap. 8. Dover Publications, Inc., 2003.
- Jarrett, R.G. A note on the intervals between coal-mining disasters. *Biometrika*, 66, 191-193.
- Kingman *Poisson Processes*. Oxford Science Publications, 1992.
- Kottas, A. Dirichlet process mixtures of beta distributions, with applications to density and intensity estimation. *In Proceedings of the Workshop on Learning with Non-parametric Bayesian Methods, 23rd ICML, Pittsburgh, PA, 2006*.
- Kottas, A., and Sanso, B. Bayesian mixture modeling for spatial Poisson process intensities, with applications to extreme value analysis. *Journal of Statistical Planning and Inference*. *Journal of Statistical Planning and Inference*, 137, 3151-3163, 2007.
- Micchelli, C.A., Xu, Y., Zhang, H. Universal Kernels. *Journal of Machine Learning Research*, 7 (2006) 2651-2667.
- Metropolis, N., Rosenbluth, A.W, Rosenbluth, M.N., Teller, A.H., and Teller, E. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 24:1087–1092, 1953.
- Mockus, J. Bayesian approach to global optimization: theory and applications. Kluwer Academic, 2013.
- Moeller, J., Syversveen, A., and Waagepetersen, R. Log-gaussian cox processes. *Scandinavian Journal of Statistics*, 1998.
- Murray, I., Adams, R.P., and MacKay, D.J.C. Elliptical slice sampling. pp. 9–16. Appearing in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), 2010.
- Pillai, N.S., Wu, Q., Liang, F., Mukherjee, S., and Wolpert, R.L. Characterizing the function space for Bayesian kernel models. *Journal of Machine Learning Research*, 8: 1769–1797, 2007.
- Quinonero-Candela, J. and Rasmussen C.E. A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research*, 6 (2005) 19391959.
- Rao, V. A. and Teh, Y. W. Gaussian process modulated renewal processes. *Neural Information Processing Systems (NIPS)*, 2011.
- Rasmussen, Carl E. and Williams, Christopher K.I. (eds.). *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- Rathbun, S.L. and Cressie, N.A.C. Asymptotic properties of estimators for the parameters of spatial inhomogeneous Poisson point processes. *Advances in Applied Probability*, 26:122–154, 1994.
- Twitter Inc. Twitter sample stream API. <https://dev.twitter.com/streaming/reference/get/statuses/sample>